

38. (Previously Presented) The method of claim 36, wherein the data events include data events pertaining to session control information.

39. (Previously Presented) The method of claim 36, further comprising the steps of:
monitoring all user actions with the host application;
creating a session monitor data set over a time period interval, wherein the session monitor data set is created by accumulating the monitored user actions into a monitoring database;
compiling the accumulated session monitor data set; and
analyzing the accumulated session monitor data set.

40. (Previously Presented) The method of claim 39, wherein the time period interval is settable by the user.

41. (Previously Presented) The method of claim 39, wherein the compiling step and the analyzing step occur at one of a scheduled time and by an invocation of the user.

42. (Previously Presented) The method of claim 36, wherein the analyzing step further comprises the step of evaluating user inputs in real-time.

43. (Previously Presented) The method of claim 36, further comprising the steps of:
separating actions involving tool bars and controls executing a single action function; and

removing these actions from the session structure.

44. (Currently Amended) The method of claim ~~35~~ 36, wherein the GuiFramework models navigational states of the host application.

45. (Previously Presented) A method of identifying logical patterns from a set of a user's inputs to a computer host application, the method comprising the steps of:

storing in a database a host application model, the model including a GuiFramework that models the host application interface controls that reflect the different states of the host application;

extracting the GuiFramework from the database and expanding the GuiFramework into an interlinked, indexed network structure in memory;

intercepting, during execution of the host application, one or more operating system messages to obtain information relating to a plurality of the user's inputs;

polling the user inputs to detect user activation of an interface device;

creating, upon detection of the user's activation, an action structure that includes status flags to mark structures;

accumulating the action structure in a database;

searching the accumulated action structures to detect sequences capable of being processed into an interactive custom accelerated procedure (ICAP); and

generating the ICAP to automate the detected sequences.

51. (Previously Presented) The method of claim 45, wherein the GuiFramework models navigational states of the host application.

52-53 Cancel